

## Rancang Bangun Sistem Kontrol ketinggian Air Bendungan Menggunakan Metode PID

**RANCANG BANGUN SISTEM KONTROL KETINGGIAN AIR BENDUNGAN MENGGUNAKAN METODE PID****Hendrawan Wahyono**S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail : [hendrawanwahyono.hw@gmail.com](mailto:hendrawanwahyono.hw@gmail.com)**Puput Wanarti Rusimamto**Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail : [puputwanarti@unesa.ac.id](mailto:puputwanarti@unesa.ac.id)**Abstrak**

Bendungan merupakan bangunan yang dibuat oleh manusia, guna menampung air sehingga terjadi genangan yang kemudian air tersebut akan digunakan untuk berbagai macam tujuan. Manfaat bendungan sangat banyak, seperti penyediaan air untuk irigasi dan untuk pengendalian banjir, irigasi, air baku atau air minum dan juga untuk pembangkit listrik serta pengembangan lokasi kawasan wisata. Sedemikian besar manfaat dari bendungan, jika dalam pengelolaan pengairan terjadi keteledoran, maka akan menimbulkan permasalahan yang besar bahkan bencana seperti yang sering terjadi saat ini yaitu banjir. Tujuan penelitian ini adalah menghasilkan dan mengetahui hasil pengujian sistem kontrol ketinggian air pada bendungan menggunakan kontroler *PID controller* sehingga dapat memperbaiki hasil respon ketinggian air *plant* miniatur bendungan agar tetap stabil sesuai nilai level air *setpoint* yang dikehendaki. Metode pengontrolan *PID controller* dirancang dengan mengidentifikasi *plant* menggunakan karakteristik respon sistem orde pertama untuk mendapatkan model matematis serta nilai logika untuk *PID* yang sesuai karakteristik respon sistem. Respon hasil pengujian sistem kontrol ketinggian air bendungan menggunakan PID lebih baik daripada sistem tanpa menggunakan kontroler di mana saat menggunakan PID, ketinggian air lebih mendekati set point (11,762 cm) daripada saat tidak menggunakan kontroler (8,78 cm), waktu yang dibutuhkan untuk mencapai set point menggunakan PID jauh lebih cepat (140 s) dibandingkan tanpa menggunakan kontroler (174 s) dan *error steady state* saat menggunakan PID kecil (1,98 %) daripada saat tidak menggunakan kontroler (32,2%).

**Kata Kunci** : Kontrol Ketinggian Air, *PID controller*, Arduino, Sensor Ultrasonic, Motor DC**Abstract**

Dams are buildings made by humans, in order to hold water so that inundation occurs then the water will be used for various purposes. The benefits of dams are very large, such as water supply for irrigation and for flood control, irrigation, raw water or drinking water and also for electricity generation and the development of tourist area locations. So much the benefit of the dam, if in the management of irrigation there is negligence, it will cause big problems and even disasters such as those that often occur today, namely flooding. The purpose of this study is to produce and find out the results of water level control system testing on the dam using a PID controller so that it can improve the results of the miniature dam plant water level response to remain stable according to the desired water level setpoint. The PID controller control method is designed by identifying the plant using the first order system response characteristics to obtain mathematical models. Response to the results of testing dam water level control system using PID is better than a system without a controller where when using PID, the water level is closer to the (11,762 cm) set point than without the controller (8,78 cm), the time need to reach the setpoint using PID is more fast (140 s) then without controller (174 s) and steady state error using PID is more closer (1,98%) than without using controller (32,2%).

**Keywords** : Water Level, *PID Controller*, Arduino, Ultrasonic Sensor, DC Motor**PENDAHULUAN**

Saat ini, banjir menjadi ancaman serius disebagian tempat dataran rendah di Indonesia. Hal ini disebabkan oleh beberapa kasus. Meningkatnya populasi manusia, semakin berkurangnya daerah serapan air akibat dari banyaknya proyek pembangunan yang kurang memperhatikan lahan hijau. Dan kurangnya kesadaran dari masyarakat mengenai pentingnya menjaga aliran air baik itu di wilayah sungai ataupun bendungan atau waduk, banyak masyarakat yang masih membuang sampah secara sembarangan sehingga menyebabkan penyumbatan sungai yang pada akhirnya dapat menyebabkan terjadinya banjir pada wilayah tersebut.

Dengan adanya masalah ini, perlu adanya suatu sistem yang dapat mengatur ketinggian air sungai atau bendungan agar tetap stabil. Sistem ini sangat diperlukan dalam menghadapi curah hujan yang tidak menentu di masing-masing wilayah. Selama ini dalam pengaturan buka tutup pintu air bendungan dan sungai masih secara manual dan masih memerlukan banyak pertimbangan sebelum dapat melaksanakan sistem tersebut. Contohnya saat hujan datang tiba-tiba pintu air di sungai pasti akan telat untuk memperbesar jalan airnya karena cuaca yang menjadi kendala manusia. Oleh sebab itu, dalam menyikapi permasalahan tersebut penulis akan memberikan solusi berupa kontrol buka tutup pintu air bendungan dan sungai secara

otomatis dengan mempertimbangkan beberapa aspek yang akan di atur agar sistem dapat bekerja dengan maksimal. Diharapkan dengan system ini nantinya dapat lebih meminimalisir peluang terjadinya meluapnya air pada bendungan karena ketinggian air akan tetap stabil bagaimanapun kondisi cuaca yang terjadi.

Arnawa menyatakan bahwa penyediaan informasi mengenai kondisi ketinggian air pada bendungan sangat di perlukan untuk persiapan wilayah sekitar dan untuk meningkatkan kewaspadaan akan terjadinya banjir, dengan memanfaatkan sensor ultrasonik untuk mendeteksi perubahan ketinggian air bendungan. Sensor ultrasonik yang digunakan memiliki kehandalan lebih baik daripada sensor yang jarak yang lain, yang dibuktikan dengan pengujian menggunakan algoritma *Markov Localization*. Selain itu dalam pemanfaatan kemampuan sensor ultrasonik pada *prototype* hanya dalam skala sentimeter, namun apabila dalam penggunaan sebenarnya maka akan disesuaikan dengan kondisi di lapangan, maka diperlukan adanya akuisisi atau *check and balance* data pembacaan nilai sensor dengan kondisi agar tidak terjadi banyak kesalahan.

Penanggulangan atau persiapan untuk menangani banjir ini sendiri memang sangat penting, karena apabila terjadi kelebihan volume air pada bendungan dan tidak segera dialirkan, hal ini dapat menyebabkan jebolnya bendungan dan pada akhirnya mengakibatkan banjir yang besar, maka dengan adanya sistem ini diharapkan kendali ketinggian air dapat lebih mudah dikendalikan secara otomatis.

Tujuan penulisan ini digunakan untuk merancang bangun sistem control ketinggian air bendungan menggunakan metode PID.

## KAJIAN PUSTAKA

### Kontroler PID

Keuntungan gabungan dari kontroler PID adalah untuk mempercepat reaksi sebuah sistem dan menghilangkan *offset*. Kontroler dengan kontrol proporsional ditambahkan dengan kontroler integral hubungannya adalah  $u(t)$  sebagai *output* dari kontroler dan  $e(t)$  adalah sinyal *error* (Ogata, 1985). Kontroler Proporsional-Integral dituliskan sebagai berikut :

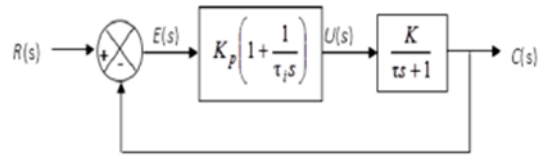
$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (1)$$

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (2)$$

Keterangan :

- $e(t)$  : sinyal *error*
- $u(t)$  : *output* kontroler
- $K_p$  : konstanta proporsional
- $K_i$  : konstanta integral
- $K_d$  : konstanta derivatif

Sebuah kontroler PID untuk kondisi *plant* orde satu membentuk sistem umpan balik dengan masukan  $R(s)$  dan keluaran  $C(s)$ , dapat digambarkan sebagai berikut :

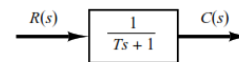


**Gambar 1** Diagram Blok Kontroler PID Plant Orde Satu

(Sumber : Ogata, 1985)

### Karakteristik Respon Orde Satu

Diagram blok sistem orde satu ditunjukkan pada gambar di bawah ini:



**Gambar 2.** Diagram Blok Sistem Orde Satu.

(Sumber : Ogata, 1985)

Secara fisik sistem ini mungkin direpresentasikan pada rangkaian RC, sistem termal, atau yang serupa. Hubungan *input-output* pada sistem orde satu dinyatakan sebagai berikut:

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1} \quad (3)$$

Keterangan:

- $C(s)$  : Sinyal *Output*
- $R(s)$  : Sinyal *Input*
- $K$  : Konstanta
- $\tau s$  : Konstanta waktu

### Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardwarenya* memiliki prosesor Atmel AVR dan *softwarenya* memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para *hobbyist* atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler.

### Matlab

Matlab adalah singkatan dari *Matrix Laboratory*, merupakan bahasa pemrograman yang dikembangkan oleh *The Mathwork Inc.* yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti *Delphi*, *Basic* maupun *C++*. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan. Pada awalnya Matlab dibuat untuk memberikan kemudahan mengakses data matrik pada proyek *linpack* dan *eispack*. Saat ini matlab memiliki ratusan fungsi yang dapat digunakan sebagai *problem solver* mulai dari *simple* sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu. (Wong, Michael. 2012)

### Sensor Ultrasonic

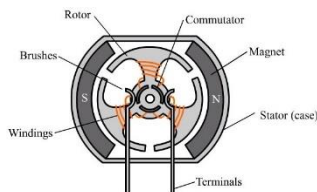
Sensor HC-SR04 adalah sensor pengukur jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini mirip dengan radar ultrasonik. Gelombang ultrasonik di pancarkan kemudian di terima balik oleh receiver ultrasonik. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor ini cocok untuk aplikasi elektronik yang memerlukan deteksi jarak termasuk untuk sensor pada robot.



**Gambar 3. Sensor Ultrasonic**  
(Sumber : Elang Sakti, 2014)

### Motor DC

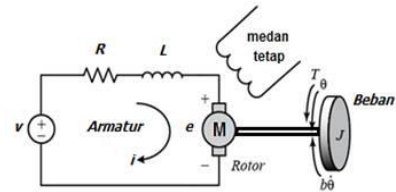
Pada motor DC kumparan medan disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran.



**Gambar 4. Bagian-bagian motor DC**  
(Sumber : Djoko Sugiono, 2015)

Pada umumnya dalam sistem kontrol banyak digunakan motor DC sebagai *actuator* pada *plant*. Secara

konsep dasar motor DC merupakan penerapan gaya Lorenz yang terjadi pada medan magnet listrik.



**Gambar 5. Rangkaian Ekvivalen Motor DC**  
(Sumber : Djoko Sugiono, 2015)

## METODE PENELITIAN

### Pendekatan Penelitian

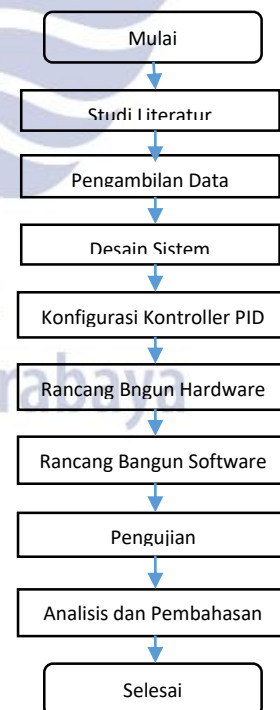
Pendekatan penelitian ini dilakukan dengan pendekatan kuantitatif dengan perhitungan angka. Pada penelitian ini menggunakan aplikasi simulasi terlebih dahulu untuk menguji coba sistem yang akan digunakan dan mengembangkan model matematis. *Software* Arduino digunakan untuk menerapkan kontroler PID.

### Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester gasal 2018/2019.

### Rancangan Penelitian

Rancangan penelitian ini dilakukan secara sistematis dan terstruktur. Tahapan penelitian ini ditunjukkan pada diagram di bawah ini :

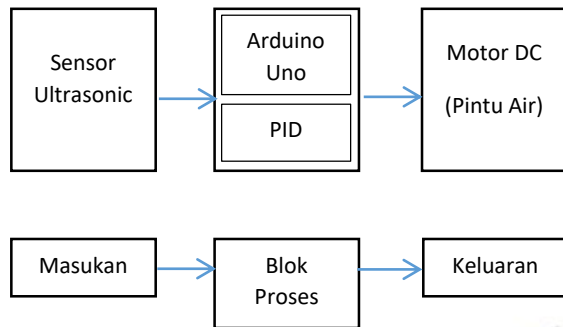


**Gambar 6. Alur Rancangan Penelitian**  
(Sumber : Data Primer, 2018)



### Desain Sistem

Desain sistem kontroler PID untuk pengendalian ketinggian air pada bendungan ditunjukkan pada gambar berikut:



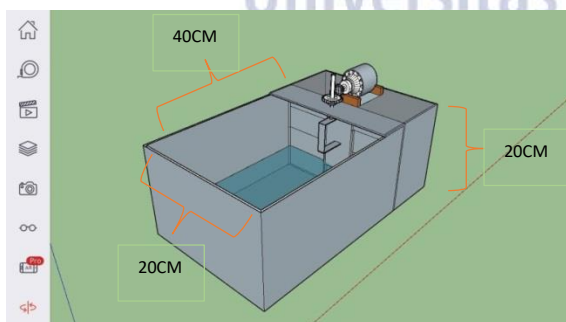
**Gambar 7.** Model Desain Sistem Kontrol  
(Sumber : Data Primer, 2018)

Pada gambar diatas merupakan desain sistem perancangan kontroler PID untuk kontrol ketinggian air yang terdiri dari blok sistem yang terintegrasi menjadi satu. Pada blok *hardware* terdiri dari *power supply*, Arduino, Sensor Ultrasonic, Driver Motor L298N dan Motor DC. *Power supply* digunakan untuk sebagai sumber tegangan listrik ke semua perangkat yang ada di dalam rangkaian dengan sumber listrik sebesar 220 volt. Sensor Ultrasonic sebagai sensor yang membaca ketinggian air pada bendungan. Dimana hasil pembacaan sensor tersebut akan dikirim ke Arduino. Arduino berfungsi sebagai alat untuk mengumpulkan atau akuisisi data. Selain itu Arduino berfungsi sebagai perantara antara laptop (PC) dan *actuator* untuk membaca nilai *input* dan *output* sistem.

Kontroler PID dan proses pengontrolannya dilakukan pada laptop (PC) dengan *software* Arduino yang nantinya akan dioalh di Matlab.

### Rancang Bangun Hardware

Rancang bangun *hardware* yang dimaksud adalah menghubungkan antar komponen, sehingga semua komponen yang digunakan dapat terhubung dan dapat beroperasi sesuai dengan desain sistem.



**Gambar 8.** Rancang Bangun  
(Sumber : Data Primer, 2018)

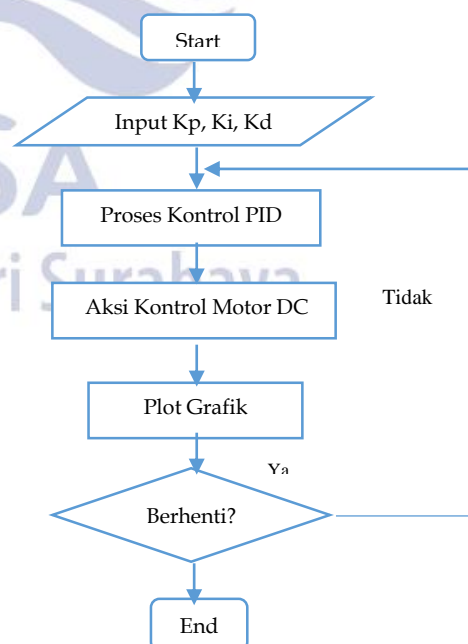
Laptop (PC) sebagai alat untuk penerima data, pemrosesan, pengiriman program ke Matlab melalui kabel USB. Untuk memonitoring kinerja sistem menggunakan *software* Matlab dengan data yang didapat dari pembacaan sensor dalam bentuk grafik atau angka. *Software* Arduino sudah terhubung dengan *Hardware* untuk proses pembacaan sensor ultrasonic yang nantinya akan menggerakkan pintu air.

### Pemodelan Sistem

Untuk mendapatkan pemodelan sistem harus mengetahui fungsi alih dari sistem. Fungsi alih didapatkan dengan cara pendekatan sistem orde-1. Sehingga untuk mendapatkan parameter-parameter dari karakteristik sistem orde-1 yaitu  $K$ , dan  $\tau$ , dilakukan pengambilan data sebanyak 4 kali dengan *setpoint* berbeda-beda. Percobaan dilakukan dengan *loop* terbuka sehingga didapatkan respon sistem orde-1. Setelah didapatkan 4 fungsi alih dari setiap *setpoint*, selanjutnya fungsi alih tersebut akan disimulasikan dan dibandingkan dengan respon sistem sesungguhnya. Setelah dibandingkan, akan didapatkan *error* antar respon sesungguhnya dan respon simulasi. Fungsi alih dengan *error* rata-rata terkecil akan digunakan sebagai fungsi alih sistem dan menjadi acuan untuk mendesain kontroler PID.

### Rancang bangun Software

*Software* yang digunakan pada penelitian ini adalah Matlab. Pada rancang bangun *software* akan dilakukan pemrograman pada *software* Matlab yang nantinya akan diintegrasikan dengan Arduino. Matlab adalah *software* pemrograman. Berbeda dengan pemrograman lainnya yang menggunakan *text*, Matlab mempunyai bahasa pemrograman berbasis grafis atau blok.



**Gambar 9.** Diagram Alir dari *Software*.  
(Sumber : Data Primer, 2018)

Dari diagram alir diatas, pengaturan ketinggian air bendungan akan selalu menyesuaikan dengan nilai *setpoint* yang diberikan. Dari pembacaan sensor akan diketahui apakah ketinggian air sudah sesuai dengan *setpoint* atau belum. Jika belum akan diketahui nilai *error*, dan nilai *error* tersebut akan diolah oleh kontroler PID. Jika nilai *error* tidak sama dengan 0 maka proses akan berulang terus-menerus, dan apabila nilai *error* sama dengan 0 atau mendekati 0 maka proses akan berhenti.

## HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas mengenai kalibrasi sensor Ultrasonic, pemodelan sistem, validasi model, perancangan kontroler PID, pengujian respon sistem.

### 1) Kalibrasi Sensor Ultrasonic.

Kalibrasi *sensor ultrasonic* dilakukan agar mendapatkan nilai level ketinggian air dalam satuan centimeter. Alat bantu kalibrasi *sensor ultrasonic* adalah penggaris manual ditunjukkan pada Tabel 1.

Tabel 1. Data kalibrasi sensor *rotary encoder* (Sumber : Data Primer, 2018)

No.	Sensor Ultrasonic	Penggaris Manual	Error
1.	6,18cm	6cm	0,18cm
2.	8,07cm	8cm	0,07cm
3.	10,11cm	10cm	0,11cm
4.	12,22cm	12cm	0,22cm
5.	14,10	14cm	0,10cm
6.	16,15	16cm	0,15cm

Dari data kalibrasi sensor *ultrasonic* yang diperoleh, dapat diketahui error rata-rata antara sensor *ultrasonic* dengan penggaris manual adalah sebesar 0,138cm.

### 2) Pemodelan Sistem

Pemodelan sistem dilakukan dengan menggunakan metode pendekatan karakteristik sistem orde-1 dan parameter yang dibutuhkan adalah K dan  $\tau$ . Metode pendekatan karakteristik orde-1 dilakukan untuk mendapatkan model atau fungsi alih sistem. Fungsi alih yang diperoleh akan digunakan untuk merancang kontroler PI dan untuk mengetahui respon sistem simulasi. Dilakukan pemodelan pada 4 *setpoint* yang berbeda, yaitu 10cm, 12cm, 14cm dan 16cm. Dan diperoleh nilai K =1,089 dan  $\tau=26,4156$ .

### 3) Validasi Sistem

Validasi model dilakukan untuk mendapatkan fungsi alih dari sistem yang memiliki respon dengan tingkat *error* terendah. fungsi alih dari 4 *setpoint* yang telah dimodelkan beserta *error* rata-rata ditunjukkan pada Tabel 2.

Tabel 2. Fungsi alih dan perhitungan *error* (Sumber : Data Primer, 2018)

<i>Setpoint</i>	Fungsi Alih	Error Rata-Rata
10cm	$G(s) = \frac{1,089}{26,415 s + 1}$	1,27%
12cm	$G(s) = \frac{0,9969}{25,815 s + 1}$	2,31%
14cm	$G(s) = \frac{0,9952}{24,944 s + 1}$	27,9%
16cm	$G(s) = \frac{0,9621}{22,622 s + 1}$	3,39%

Dari tabel 2, dapat diketahui G(s) atau fungsi alih dengan nilai *error* rata-rata terendah adalah fungsi alih dari *setpoint* 10 cm yang memiliki *error* rata-rata sebesar 1,27 %. Fungsi alih dengan *setpoint* 10 cm selanjutnya akan digunakan sebagai acuan untuk menentukan parameter dari kontroler PID. Perhitungan parameter respon riil secara *open loop* dengan *setpoint* 10cm dan nilai  $\tau=1,089$  ditunjukkan pada Tabel 3.

Tabel 3. Parameter Respon *Plant Riil* dengan *setpoint* 10cm (Sumber : Data Primer, 2018)

Parameter	Nilai
Waktu tunak (0,5 %)	132,07 detik
Waktu tunak (2%)	105,66 detik
Waktu tunak (5%)	79,245 detik
Waktu naik (5%-95%)	77,773 detik
Waktu naik (10%-90%)	77,778 detik
Waktu tunda ( $\tau_d$ )	18,31 detik
Presentase <i>Error Steady State</i>	14,7%

### 4) Perancangan Kontroler PID

Perancangan kontroler PID dilakukan untuk mendapatkan parameter dari kontroler PID yaitu Kp dan Ki. Sudah ditentukan sebelumnya bahwa fungsi alih dengan *setpoint* 10cm memiliki *error* rata-rata terendah dan dijadikan sebagai acuan untuk menentukan parameter kontroler PID, berdasarkan persamaan 3 didapatkan sebagai berikut :

Fungsi alih *setpoint* 10cm.

$$G(s) = \frac{1,089}{26,415 s + 1}$$

Diketahui parameter :

$$K = 1,089$$

$$\tau = 26,415$$

Pada perancangan kontroler PID ini, penulis menginginkan sistem agar dapat 2x lebih cepat. Maka nilai  $\tau^*$  adalah :

$$\tau^* = \tau \times 0,5$$

$$\tau^* = 1,089 \times 0,5$$

$$\tau^* = 0,5445$$

Selanjutnya untuk menentukan nilai  $K_p$  dan  $K_i$ , dimana diketahui nilai  $\tau = \tau_i = \tau_d = 0,34658$ . Nilai  $K_p$  dan  $K_i$  adalah :

$$K_p = \frac{\tau_i}{\tau^* \cdot K} \quad (4)$$

$$K_p = 1,861$$

$$K_i = \frac{K_p}{\tau_i} \quad (5)$$

$$K_i = 0,070456$$

$$K_d = K_p \times \tau_d = 49,158 \quad (6)$$

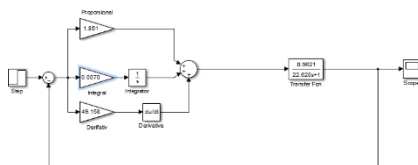
Tabel 4. Parameter Kontroler PI  
(Sumber : Data Primer, 2018)

Parameter	Nilai
$K_p$	1,861
$K_i$	0,070456
$K_d$	49,158

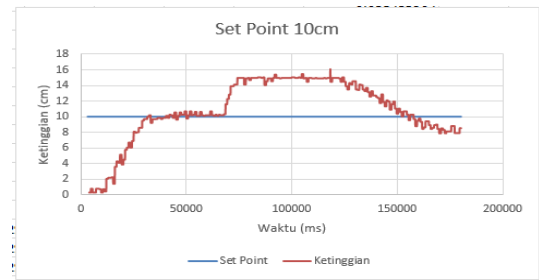
## 5) Implementasi Kontroler PID dan Pengujian Respon Sistem

Setelah didapatkan nilai  $K_p$ ,  $K_i$  dan  $K_d$ , dilakukan implementasi kontroler PID terhadap *plant*. Sebelum mengimplementasikan pada riil *plant*, terlebih dahulu disimulasikan dengan menggunakan *software* Matlab 2014.

Fungsi alih yang digunakan adalah fungsi alih dengan *setpoint* 10 cm yang telah didapatkan dari pemodelan sebelumnya. Dilakukan simulasi untuk mengetahui pengaruh kontroler PID pada fungsi alih dengan *setpoint* 10 cm. Diagram blok Matlab akan ditunjukkan pada Gambar 10 dan hasil simulasi respon sistem akan ditunjukkan pada Gambar 11.

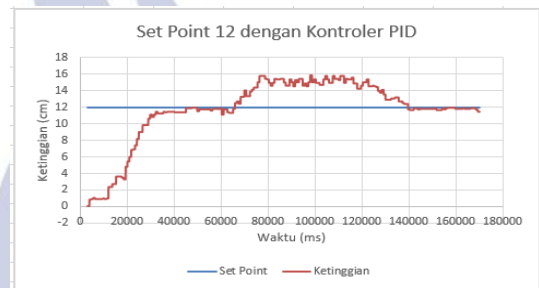


Gambar 10. Diagram Blok pada Simulink Matlab (Sumber : Matlab, 2014)

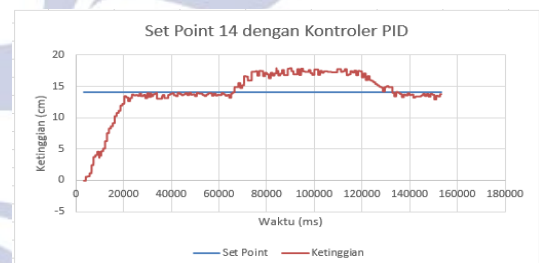


Gambar 11. Grafik Respon Simulasi Fungsi Alih 10cm dengan Kontroler (Sumber : Data Primer, 2018)

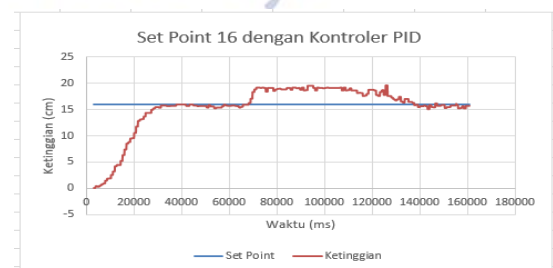
Pengujian respon pada riil *plant* menggunakan empat metode yaitu dengan *setpoint* tetap. Pengujian ini dilakukan pada tiga *setpoint* berbeda yaitu 12 cm, 14 cm dan 16 cm. Hasil respon kontroler PID pada riil *plant* dengan *setpoint* 12 cm, 14 cm dan 16 cm akan ditunjukkan pada Gambar 12, 13, dan 14.



Gambar 12. Respon Kontroler PID dengan *Setpoint* 12 cm  
(Sumber : Dokumen Pribadi, 2018)



Gambar 13. Respon Kontroler PID dengan *Setpoint* 14 cm  
(Sumber : Dokumen Pribadi, 2018)



Gambar 14.. Respon Kontroler PI dengan *Setpoint* 16 cm  
(Sumber : Dokumen Pribadi, 2018)



Hasil perhitungan parameter respon sistem dengan menggunakan kontroler PID pada *setpoint* 12 cm ditunjukkan pada Tabel 5, dan diketahui nilai  $\tau = 23,282$

Tabel 5. Parameter respon sistem menggunakan kontroler PID dengan *setpoint* 12 cm. (Sumber : Data Primer, 2018)

Parameter	Nilai
Waktu tunak (0,5 %)	116,4 detik
Waktu tunak (2%)	92,12 detik
Waktu tunak (5%)	69,84 detik
Waktu naik (5%-95%)	68,55 detik
Waktu naik (10%-90%)	51,15 detik
Waktu tunda ( $\tau d$ )	16,13 detik
Presentase Error Steady State	1,98 %

Hasil perhitungan parameter respon sistem dengan menggunakan kontroler PID pada *setpoint* 14 cm ditunjukkan pada Tabel 6, dan diketahui nilai  $\tau = 18,248$

Tabel 6. Parameter respon sistem menggunakan kontroler PID dengan *setpoint* 14 cm. (Sumber : Data Primer, 2018)

Parameter	Nilai
Waktu tunak (0,5 %)	91,24 detik
Waktu tunak (2%)	72,99 detik
Waktu tunak (5%)	54,74 detik
Waktu naik (5%-95%)	53,73 detik
Waktu naik (10%-90%)	40,09 detik
Waktu tunda ( $\tau d$ )	12,64 detik
Presentase Error Steady State	3,9 %

Hasil perhitungan parameter respon sistem dengan menggunakan kontroler PID pada *setpoint* 16 cm ditunjukkan pada Tabel 7, dan diketahui nilai  $\tau = 20,02$

Tabel 7. Parameter respon sistem menggunakan kontroler PID dengan *setpoint* 16 cm. (Sumber : Data Primer, 2018)

Parameter	Nilai
Waktu tunak (0,5 %)	100,1 detik
Waktu tunak (2%)	80,08 detik
Waktu tunak (5%)	60,06 detik
Waktu naik (5%-95%)	58,95 detik
Waktu naik (10%-90%)	43,99 detik
Waktu tunda ( $\tau d$ )	13,87 detik
Presentase Error Steady State	4,4 %

Perbandingan hasil pengujian Yss (nilai *steady state*), Ess (*error steady state*) dan Wss (waktu yang dibutuhkan untuk mencapai *steady state*) data tanpa *controller* dan menggunakan *controller* terdapat pada tabel 8 sebagai berikut.

Tabel 8. Perbandingan respon system. (Sumber : Data Primer, 2018)

Parameter	Tanpa Controller			Controller PID		
	12cm	14cm	16cm	12cm	14cm	16cm
Yss (cm)	8,78	10,25	13,9	11,76	13,45	15,56
Ess (%)	32,2	37,5	21	1,98	3,9	4,4
Wss (s)	174	180	183	140	143	153

## PENUTUP

### Simpulan

Alat sistem kontrol ketinggian air bendungan berbasis PID telah berhasil didesain dan dihasilkan menggunakan sensor ultrasonic HC-SR04 sebagai pendeteksi ketinggian air di dalam *prototype* bendungan, arduino uno sebagai mikrokontroler, serta motor DC sebagai actuator penggerak pintu air bendungan.

Respon hasil pengujian sistem kontrol ketinggian air bendungan menggunakan PID lebih baik daripada sistem tanpa menggunakan kontroler di mana saat menggunakan PID, ketinggian air lebih mendekati set point (11,762 cm) daripada saat tidak menggunakan kontroler (8,78 cm), waktu yang dibutuhkan untuk mencapai set point menggunakan PID jauh lebih cepat (140 s) dibandingkan tanpa menggunakan kontroler (174 s) dan *error steady state* saat menggunakan PID kecil (1,98 %) daripada saat tidak menggunakan kontroler (32,2%). Data diatas dapat dilihat pada tabel 8.

### Saran

Berdasarkan hasil penelitian yang telah dilakukan terdapat beberapa aspek yang dapat diperbaiki dan dikembangkan agar memberikan hasil yang lebih maksimal.

Model atau fungsi alih yang didapatkan masih kurang baik, sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini.

Menggunakan sensor *Ultrasonic SRF04* yang lebih akurat dan presisi agar respon kecepatan yang didapat lebih baik.

Sistem ini masih dapat dikembangkan dengan menggunakan metode kontroler yang lain seperti *fuzzy logic*, *model predictive control* (MPC), dan *Linear Quadratic Regulator control* (LQR) untuk mendapatkan hasil respon kecepatan yang lebih baik.

### DAFTAR PUSTAKA

- Alfatah, Muhammad Rasyid. 2016. *Prototype Sistem Buka Tutup Otomatis pada Pintu Air Bendungan untuk Mengatur Ketinggian Air Berbasis Ariduino*. Universitas Muhamadiyah Surakarta.
- Sakti, Elang. 2014. *Cara Kerja Sensor Ultrasonic, Rangkaian dan Aplikasinya* (Online). ([www.elangsakti.com/2015/05/sensor-ultrasonik.html](http://www.elangsakti.com/2015/05/sensor-ultrasonik.html) diakses pada tanggal 6 Mei 2018).
- Ogata, Katshuhiko. 1985. *Teknik Kontrol Automatik Jilid 1*. Terjemahan Edi laksono. Jakarta : Erlangga.
- Sugiono, Djoko. 2015. *Model Matematika Motor DC*. (Online). ([www.vedmalang.com](http://www.vedmalang.com) diakses pada tanggal 6 Mei 2018).
- Michael, Wong. 2012. *Pengenalan Matlab pada Sistem Kontrol*. (Online). ([www.wongmichael.wordpress.com/2012/10/21/178/](http://www.wongmichael.wordpress.com/2012/10/21/178/) diakses pada tanggal 6 Mei 2018).